The Report Committee for Miguel Angel Otero Lanuza
Certifies that this is the approved version of the following report:

Rationale Management as the Basis of Knowledge Preservation for Enterprise Systems Value Added Resellers

APPROVED BY
SUPERVISING COMMITTEE:

Supervisor:

K. Suzanne Barber

Thomas Graser

Rationale Management as the Basis of Knowledge Preservation for Enterprise Systems Value-Added Resellers

by

Miguel Angel Otero Lanuza, B.S.

Report

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Master of Science in Engineering

The University of Texas at Austin

December 2010

Dedication

Dedicated to Tatiana, Valentina, and Mariana for all their love and support during these two and a half years. This report as everything else I do is inspired by you.

Acknowledgements

I would like to thank my company, Nextec Group, its owners Eric Frank, Russell Harper, and Alan Subel for their encouragement, confidence, and financial backing to continue with my education. Also would like to thank my supervisors, Chuck Oliver and Daniel Cattarusa, for their recommendations when I applied to this program and complete support during these years.

I would like to thank my supervisor, Dr. Suzanne Barber, and my reader, Dr. Thomas Graser, for their advice and feedback.

I would also like to express my gratitude to the three participants in the surveys presented in this paper for providing their valuable insight about this topic.

Lastly, my appreciation to my classmates Ryan Brown and Bruce Zabava for assisting me with reviewing this report and providing suggestions.

Abstract

Rationale Management as the Basis of Knowledge Preservation for Enterprise Systems Value Added Resellers

Miguel Angel Otero Lanuza, M.S.E

The University of Texas at Austin, 2010

Supervisor: K. Suzanne Barber

Enterprise systems (ES) implementation, especially Enterprise Resource Planning systems (ERP), is an extensively researched topic in recent years. Existing papers focus mainly on the success or failure of the project analyzed from the client's standpoint. Although authors agree that a successful implementation requires the participation of consultants from a Value-Added Reseller (VAR), little or no work has been published that examines the topic from this perspective. While it is true that this kind of implementation is not strictly related to the traditional software development lifecycle,

the two have many things in common and the former can benefit from software engineering techniques. Intellectual capital found in the heads of consultants, developers, project managers, and all other project stakeholders is VAR's main asset as well as in most of software-related organizations. Hence, it is critical to preserve it in order to safeguard the foundation of these organizations. The goal of this paper is to propose rationale management as the basis of knowledge preservation for enterprise systems VARs. Enterprise systems implementation process, including its actors, challenges, and the knowledge that surrounds it, is examined to justify the proposal. To assess the perception of real-world VARs about knowledge management applicability and their existing strategies, a questionnaire was applied to 3 executives. Their answers confirmed that knowledge is considered vital to their organizations but the methodologies as well as the tools currently utilized to preserve it are rudimentary and distant from the theoretical literature.

Table of Contents

List of Figures

Introduction

Research in the field of implementation of enterprise systems (ES), especially Enterprise Resource Planning Systems (ERP), is vast. According to [1], the number of papers published from 2000 to 2006 is 125. In [1] and [2] this research topic is divided into five categories: case study, critical success factors, change management, implementation process, and cultural issues. In spite of the great interest shown, all of the research found is focused in the ERP implementation only from the client's perspective. One of the critical success factors discussed in the aforementioned papers is that a knowledgeable partner from the ERP's vendor is necessary to achieve a successful implementation. The services that partners provide add value to the implementation, therefore the name Value Added Reseller (VAR). A VAR that is in total control of its most important asset, knowledge, can provide the most value to the client during the implementation. In consequence, finding an efficient way to capture and retrieve knowledge can represent a significant return on investment for the VAR.

This paper looks into the use of rationale management, a software engineering technique intended to preserve alternatives considered about a design decision [37]. Its use could potentially be an important aid to safeguard knowledge and contribute to the success of VARs by means of exploiting it for training, maintenance, process improvement, and reference. The type of knowledge that is generated from ERP implementations is very similar to the knowledge preserved by rationale management techniques. During implementations, a VAR not only provides technical expertise to deploy, install and configure ERPs but also with the knowledge to adapt clients' business processes to the best practices materialized in the system or customize the ERP. For example, even though no two implementations are the same, there exist a high number of

tasks that can be considered repetitive along with problematic scenarios that were successfully resolved by a consultant in previous implementations. By preserving not only the final decision that is documented as part of the project's library, but also the alternatives the consultant evaluated, it would provide critical knowledge for future implementations that otherwise would only reside in the consultants' head.

This paper proposes the use of rationale management as the fundamental knowledge preservation technique for VARs, and introduces the challenges and considerations associated with this technique. From existing literature it is clear that despite of being a well researched and promising field, rationale management still need further refinement before it can be widely adopted. The main reason is it has not permeated well into software-related organizations due to the lack of adequate tools that make its use viable without imposing an enormous amount of overhead. To contrast the claims made in this paper with the opinion of the industry, a survey was applied to three top executives from ERP's VARs. From their answers, the use of knowledge in their organizations and their opinion on this subject confirms the potential benefits that rationale management could yield. The main contribution of this paper is aimed to consider these challenges and opinions as feedback of the aspects that should be incorporated into future research.

The remainder of this paper is structured as follows. In section 2, ERP implementations are analyzed from the VAR's perspective to put in context the requirements for knowledge preservation techniques. In Sections 3 and 4, knowledge and rationale management theory and their relation with software engineering is described in detail. Section 5, describes the proposed application of rationale management and the questionnaire results are presented. Finally, conclusions are drawn in Section 6.

Enterprise Systems Implementations

An Enterprise Resource Planning System (ERP) is a type of enterprise information system that integrates the majority of an organization's business functions under a uniform software environment and logical database. ERPs have become the dominant information system in use by every organization from midsize to large companies [38]. An ERP standardizes best business practices by incorporating them into the processes they implement within and across its modules. An ERP guarantees data consistency and enforces business' rules for all entities involved in transactions in contrast with islands of information that typified old legacy systems. The most important characteristics that can be found in commercial ERPs in the market are: modularity, process-orientation, and integrative.

Modularity

ERP systems incorporate multiple modules into a shared framework and standardized software environment, enabling seamless integration among each other and encompassing complete business processes of an organization. One example of this is the purchasing process: a requisition is created in the Requisitions module; when approved a purchase order is generated and later the items ordered are received in the Purchasing module; the receipt operation affects available quantities in the Inventory module; further a voucher will be the originating document for a check to pay for the items and complete the process. All these modules are separate entities themselves and can be utilized by different departments of an organization, but under an ERP umbrella they are transparent processes to the user that guarantee consistency in the information without requiring separate data entries.

Process orientation

As mentioned in the previous section, ERPs functionality reflect the way an organization operates not only from an internal department's perspective but cutting across several organizational areas. Seamless connections between modules and a uniquely shared database allow users to see the same up-to-date information. Traditionally, legacy systems were designed for a specific department in mind resulting in multiple disconnected systems that supported an organization's operations. Also, these systems processed and stored data internally maintaining their own copies of it so communication with different systems that were used by other departments was done through batch processing. For this reason, multiple interfaces were required, as many as the number of departments or systems a software package interacted with. In addition, implement real-time processing was impossible since data was allowed to affect other systems only when a batch was created, exported and then imported by the receiving system.

Integrative

By means of maintaining a single logical database, an ERP enables organizations to ensure that the information seen by all the company at any given time is unique and current, precluding the islands of information phenomenon. In addition, standardization of the user interface as well as functionality across all the modules simplifies training, processing and access to information. The possibility of customizing and extending the functionality of the ERP enables implementers to close the gaps between the organization's requirements and the ERP's standard capabilities. By developing new custom modules inside of the ERP environment, the organization minimizes the number of systems that exist outside of the ERP

integrating all of its information under one roof and taking advantage of its benefits.

RESEARCH BACKGROUND

The amount of research generated in recent years in the topic of ERP's implementations is vast and diverse mainly due to its multidisciplinary characteristics and wide acceptance from practically every medium to big size companies [1]. Literature surveys on the topic [1, 2] agree about the classification of ERP publications categorizing them by: Implementation, Optimization, Management, ERP software, ERP for Supply Chain Management, and Case Studies. In [1], Moon reports having reviewed 313 articles published between 2000 and 2006. The majority of these publications, over 40%, were classified as Implementation of ERPs, proving that this subtopic is the one that has caught the greatest attention among the research community.

Use of Knowledge Management Strategies during and after ERP implementations has been studied in more than a few publications. Knowledge sharing has been mentioned [3, 4] as the most important goal not only to achieve a successful implementation but to gain a true competitive advantage from the use of the ERP. In [3], the challenges that pose transferring knowledge between the main ERP actors involved in an implementation, the client, vendor, and VAR, and how knowledge management can support these flows are presented. Leknes et al. [3] studied an ERP implementation interviewing the aforementioned actors attempting to answer the question "*What role can knowledge management play in supporting the implementation of ERP systems?* The conclusions presented confirm that there exists interest in the potential benefits of using knowledge management. According to their findings, in order to achieve efficient communication to overcome knowledge barriers between partners and clients the role identified should be "*coordinate information flows and the interaction between people*

and technology…" In another publication, knowledge sharing by internal ERP implementation teams is analyzed from eight organizational culture dimensions [4]. Jones et al. studied multiple implementation cases also indicating, like in the previous research, that a critical success factor is breaking down knowledge sharing obstacles.

According to [5], in order to obtain a competitive advantage from an ERP implementation, it is critical that organizations are able to successfully manage knowledge sharing from the beginning instead of expecting that the intrinsic nature of an ERP system will enhance it. Vandaie [5] claims that even though a successful implementation is the first objective of every implementation, the ultimate goal of using an ERP is to obtain a competitive advantage in spite of the potential possibility of competitors using the same business practices imposed by it. In his work, from a literature research, two areas of concern for ERP knowledge management are identified: tacit knowledge and issues with respect to the process-based nature of organizational knowledge. To mitigate these concern areas and maintain the competitive advantage, it is important to successfully internalize and integrate the adopted processes into the organization through an effective knowledge sharing and transfer between the ERP internal team and members of the organization as well as with external consultants.

From these findings, it is clear that there is agreement in considering knowledge management and sharing a very important factor for the success of ERP implementations and posterior use. As mentioned before, one of the principal interests has been to dissect the critical success or failure factors that determine the outcome of an ERP implementation but only including internal aspects that the client needs to do or avoid. It is also recognized [5] that a successful implementation must have a vendor's partner that knows not only how to use ERP but the best way to take advantage of it depending of the particularities of the client's requirements. Despite of this acceptance, it is remarkable

that all of the researched papers approach the implementation of ERPs only from the client's standpoint without considering the management of knowledge in the VAR's context. Considering the fact that knowledge management is directly associated with successful implementations and the main source of knowledge about the implementation is the VAR, it is noteworthy not finding an article that digs further into the critical role that knowledge preservation plays for them.

VALUE ADDED RESELLERS

Today, it is uncommon to find organizations that decide to develop or even implement an ERP by themselves. Because of their high complexity nature, the majority of these IT projects are carried out by the acquisition of existing commercial systems and its implementation is done with the guidance of the vendor's partner or Value-Added Reseller (VAR).

VAR's Activities

In general, VAR's provide with consulting services that assist organizations implementing ERPs in many ways:

- Finding the most suitable ERP system that fits the client's business model's requirements.
- Technical support to review and suggest the optimal configuration of the hardware that will host the system.
- Installation and configuration of the ERP system according to the vendor's guidelines and client's requirements and business nature.
- Assistance with the change management associated with new processes.
- End-user and administrators training.

- Customizations to adapt the standard ERP's functionality to client's business processes' requirements.
- Migration of data from old systems to the new one and development of interfaces to import or export data to or from external systems.
- Guidance on change management to cope with the adaptation of client's processes introduced by ERP's "best practices".

VAR's actors

A typical implementation of an ERP by a VAR involves the participation of consultants with different competencies that carry out the installation, setup, and customization tasks [5, 6]. A common categorization according to the role the consultant plays during the implementation, classifies them in project manager, technical consultant, application consultant, and developer.

Project Manager

A project manager is responsible for supervising the overall and detailed progress of the implementation. Planning, organizing, selecting consultants, monitoring, and managing the project are the main activities of a VAR's project manager. Scheduling and planning the work breakdown structure according to the project's goals is one of the most important activities he carries out because this will be the benchmark against which the success of the project will be measured. Once the schedule is established, he assigns tasks to the most suitable consultant depending of its complexity and characteristics to assure they are finished according to the objectives. On the other hand, an effective control of the project through frequent monitoring and feedback is a fundamental task a project manager performs. Supervising risk to avoid deviations from the initial budget and

schedule will allow the manager to take appropriate measures to mitigate those risks, avoiding potential problems that could jeopardize the successful outcome of the project.

Application Consultants

Application consultants in the majority of the implementations are in charge of leading the project and be responsible for the strategic decisions. In general, an application consultant will determine how to properly setup the system in order to address the gaps between the client's requirements and the standard functionality. Depending on the size of the implementation a single consultant can perform more than one activity but it could turn into more than one consultant working on them. Also, because of the many-sided nature of an ERP it is hard to find application consultants that are experts in all the business processes that modules implement. For example, consultants with an accounting profile will be responsible for the implementation of financial modules while industrial engineers will implement manufacturing ones.

One of the key aspects application consultants deal with is determining the best way to implement the ERP by addressing the differences between the business requirement and the standard functionality. For this reason, it is expected from them to have a deep understanding of the capabilities of the ERP system as well as the client's requirements in order to decide if a customization is required or if it is possible.

Developers

Developers are responsible for customizing the ERP, extend its functionality, and integrate it with other systems by creating interfaces to import or export data. They are involved in design decisions to provide their expertise on customizations to determine if they can be done, and the estimate of their duration. It is also common for developers to participate in the initial load of data to transform and migrate it from former systems.

Technical Consultants

Technical consultants generally carry out the installation of the system in the client's environment. Part of the deployment of the ERP involves creation of databases, technical configuration, and installation of servers and clients. It is also common for technical consultants to create test environments for testing and training purposes. In addition, sometimes it is required to perform an evaluation and assessment of the hardware environment and make suggestions about the architecture in order to optimize the performance of the system.

VAR's Challenges

As mentioned before, the initial goal of an ERP implementation is to deliver the system according to the planned budget and schedule. From the client's point of view, in the long-term perspective additional objectives include achieving a competitive advantage and a return of the investment through making the business operations more efficient. Being the VAR a critical piece whose performance directly impacts the success or failure of implementations, understanding the key factors that enable it to repetitively provide clients with valuable assistance is a meaningful contribution to the study of ERP systems.

It is inaccurate to consider ERP implementations identical although it can be argued that they follow the same procedures and guidelines indicated by the system's vendor. It is true that all clients using the same ERP's version will end up using the same system including its main core functionality (with the exception of the customizations applied). However, no two implementations are the same and this means for VARs that on top of the complexity involved in a single implementation, they have to face the same complexity factor along with its countless variables multiplied by the number of clients they serve.

All the activities that entail the consulting services that VAR's actors perform during ERP implementation lifecycle are highly intellectual with knowledge playing a critical role. Knowledge about implementations can take many forms and be stored in different repositories but, the most common place where it resides is in consultants' heads. Being knowledge the most important asset a VAR possess it is risky in more than one way to allow knowledge to be located in such a volatile location. This is the main challenge a VAR faces because knowledge not only can walk out the door when a consultant leaves but also it can be forgotten as time passes by. In addition, another challenge is how to effectively transfer knowledge among consultants to be used when it is required for training, providing support, during future upgrades, or making modifications to customizations.

At the same time, although preserving knowledge for VARs can be considered a critical piece of their business model, the overhead imposed discourages these initiatives. When weighing the option between adding additional documentation to capture knowledge about implementations and preserve only the essential, the majority of VARs will rather choose the latter. This commonly occurs because of the fact that to spend documenting a significant number of hours cannot be directly charged to the client because it is not perceived as added value to the implementation.

ERP IMPLEMENTATIONS

Critical Factors

One of the most studied aspects of ERP implementations is the analysis of the critical factors that contribute or hamper success in these projects. In [1], 15 published articles are identified that analyze lessons learned from previous implementations and the

reasons that contributed to their success. A summary of the most relevant factors mentioned is listed below:

- **Top management support.** A strong commitment from top executives has been verified as one of the most important success factors. The ERP implementation project needs to be identified *"publicly and explicitly as top priority"* [10].
- **Clear business plan and vision.** After two decades of ERP implementations it has been proved that these projects should be driven by a strategic business vision rather than pure technicalobjectives [12]. As Davenport [12] states, fundamental decisions about ERP's implementations should be approached from a business perspective because of the amount of business change and cost involved. For this reason, when putting together the vision of the project, firms must think forward about what the organization will need in the future. Ultimately, the final decision should be communicated to the organization in business terms rather than technical.
- **Change management.** An ERP implementation by definition implies changes to business processes [11]. Reaction to change is a normal response in users and originates in employees that feel their status quo is being threatened by the changes introduced. An adequate management to introduce these changes along with the management readiness to change and the scope of the change all together contribute to the success of a change management strategy [10].
- **Help of consultants.** An implementation team is composed by a mixed balance of internal stakeholders from different areas and external

consultants from a VAR. Selecting the consulting firm that can provide their expertise and know-how in implementation methodologies is also part of successful projects [10]. Working closely with consultants during all implementation phases is critical in order to obtain the knowledge required to continue the operation of the ERP system after the VAR leaves [3].

It is important to mention that the majority of the critical factors identified are closely related to internal factors that the client needs to keep in mind in order to accomplish a successful implementation. However, there is also agreement about the crucial roles VAR's consultants play in every implementation. This implies that expectations about their participation are high and even though the basis of success leans more towards the client, the guidance a VAR provides will steer the project in the right direction. For this reason, a VAR that can offer their services in a timely manner, with efficient use of resources, and deliver a solution according to the plan, has an advantage that can only be obtained if they are able to preserve knowledge and use it efficiently when needed. The following sections will analyze the role of knowledge in VARs and advantages and disadvantages of available methodologies to provide with better services.

Phases

The categorization proposed in [8] for all the stages an ERP implementation goes through includes the following phases: Adoption decision, Acquisition, Implementation Use and Maintenance, Evolution, and Retirement. During each phase, VARs participate in different ways, and even in some cases, depending of the level of expertise required, more than one VAR can take part of different phases.

Adoption decision

This is the phase that initiates every ERP implementation project when organizations decide they need to replace or upgrade their existing systems. General requirements and objectives are defined according to the analysis of the benefits and impact in the organization. In the majority of the implementations, this is a phase in which VARs do not have a great level of participation, unless they assist with analyzing the current status of the organization and the potential benefits of replacing current systems with an ERP. During the decision processes an organization goes throughout in this phase, it is important to stress that an implementation of an ERP is not only an IT project. The changes imposed by an ERP to organizational operations can be extraordinary because of its integrated nature in comparison with traditional systems [7].

Acquisition

During this phase VARs assist clients with selecting the most suitable ERP that fits the organization's requirements. Even though all ERPs are, by definition, integral they can be oriented towards different types of business. Throughout the acquisition phase, fundamental knowledge about the implementation is elicited since a critical decision for the success of the implementation is selecting the correct ERP [9]. Among the decisions that support the implementation and are made when selecting the ERP include: budget, time-frame, goals, and deliverables [5].

Implementation

Typically, the heaviest participation of VARs is during this phase in which the ERP implementation is designed, parameterized, customized, and deployed according to the requirements established in the previous phase. In addition, training and knowledge transfer occur to prepare users to be ready to use the system when it goes live. Guided by

requirements, knowledge generated from this phase constitutes the body of any knowledge preservation initiative since consultant's previous experience along with vendor's implementation guidelines are applied to meet the needs of the client. In contrast with a traditional system development, ERP's main focus is on a business process design rather than technical analysis and programming. The exception is the extension of standard functionality that, even though can be considered a separate software development project, its aim is still oriented at the main business goals of the implementation.

Use and Maintenance

During the use and maintenance phase, the ERP is expected to yield its benefits while the level of adaptation of the organization to its processes tightens. For VARs, supporting implementations after they are deployed in production environment turns into a challenge as consultants start to work in new projects and knowledge about the implementation begins to fade out if it was not preserved. Also, regularly maintenance activities and unexpected events that require corrections are part of the support tasks. These can turn problematic if different consultants than the ones that were responsible for the implementation are in charge of support and necessary knowledge is not accessible.

Evolution

Like with every software development or IT project, the scope that delimits what will be included in the implementation must be determined in earlier phases. When the ERP is used in a production environment, many new requirements arise from opportunities that either did not exist before or was not possible to realize them because of former systems' limitations. According to [8], two types of extensions to integrate new functionality to the ERP can be classified in: *"Upwards"* and *"outward"*. An *"upward"*

extension includes those that make advanced use of the data the ERP generates like data warehouses, and business intelligence systems. The *"outward"* type comprises the integration of new modules or interface with external systems like customer relationship management, or electronic commerce. For the VAR, any extension requires their participation and to share the knowledge about the implementation. In some cases, the same VAR that lead the implementation can extend the functionality if they have the required competencies, but in others, a new consulting firm is hired by the client to be in charge. In both cases, knowledge preserved is fundamental to successfully evolve the ERP.

Retirement

In the final phase of the lifecycle, the ERP is replaced due to new business' needs or as new technologies emerge. Most of the time, if the results were as expected and the investment returned according to the objectives, the client will go through repetitive cycles that will upgrade the same ERP to a new version. In this case, for a VAR, being able to keep its clients and support them throughout these cycles guarantees continuous business growth. The main advantage in this situation is that the knowledge required is already preserved and can be employed to simplify upgrades. Otherwise, for a client it is an easier decision to consider replacing the VAR in cases when the same ERP will only be upgraded.

Knowledge Management Concepts

This section presents a brief summary of knowledge management theoretical concepts. As described in [35], knowledge can be easier to describe by distinguishing it from data and information. Data are discrete facts without an inherent meaning, like a number, name, etc. Information can be seen as data that has been transformed by adding it value so it is contextualized, categorized, calculated, corrected or condensed. Finally, even though knowledge can have many different meanings, from its characteristics it can be defined as "*mix of framed experience, values, contextual information, and expert insight that provides a framework for evaluating and incorporating new experiences and information*".

A generic definition for knowledge management states that it is a discipline that provides methods to manage the creation, preservation, and application of the knowledge that is available within an organization [32]. In the context of software engineering, support to software development activities by managing knowledge involved is not a new problem. In [33] an initial attempt described the concept of *experience factory* as a way to represent and store in an *experience base* experiences from software development projects so they can be reused.

KNOWLEDGE REPRESENTATION

Knowledge representation can be seen from two perspectives: *formal* and *informal*. On one hand, formal representations facilitate automatic processing and retrieval of knowledge, a quality that is important since the amount of knowledge generated in software engineering activities grows very quickly and its complexity can be unmanageable. To solve this problem, knowledge engineering is a discipline that attempts to formalize knowledge methods in order to build knowledge-based systems

whose objective is to standardize its representation by creating models of it [30]. On the other hand, an informal representation provides with an easier way of codification and accessibility when utilized by humans but its lack of structure make it difficult to index and store by knowledge based systems [28]. Informal representations can include emails, notebooks, recordings of meetings, etc.

KNOWLEDGE CLASSIFICATION

Knowledge can take many different forms and types and be transformed from one type into another. A general classification divides knowledge into explicit knowledge and tacit knowledge. Explicit knowledge is the kind of knowledge that is represented in a sensible form like words, formulas, numbers, etc. and is captured in a medium to be transmitted like documents, audio, video, etc. On the opposite, tacit knowledge is a combination of explicit knowledge internalized with previous experiences of its possessor that resides in his head. To be transmitted, it needs to be de-codified into an explicit form that will then be codified by the receiver into its own mental model. The ability to transmit knowledge is crucial, since by sharing and exchanging it more knowledge is generated preventing it to become obsolete and worthless [29]. In VAR's context, explicit knowledge is easy to find in knowledge bases provided by the ERP vendor or built by the VAR itself. On the contrary, tacit knowledge is difficult to elicit because of its characteristics and the lack of models or tools that simplify its preservation and retrieval.

KNOWLEDGE FLOWS

One of the objectives of knowledge management is to enhance its flow across organizations to guarantee it can be used when needed [34]. In software engineering companies, the flow of knowledge is normally transferred through documentation like between developers and maintainers. Some of the most important problems of software

related companies are related to an inadequate knowledge flow. Two causes identified in [34] are the failure to maintain up-to-date documentation and personnel turnover that either leaves the company or is assigned to new projects. Therefore, to support software organizations, knowledge management initiatives need to identify activities, knowledge required and generated, and the channels where knowledge will flow through. An important question that drives this type of research is how to take one step further beyond the usual documentation approach, which only records the final outputs from experiences, by including rationale [28].

KNOWLEDGE MANAGEMENT APPROACHES

Depending on the formalization level, [31] classifies approaches for knowledge management in two: codification and personalization strategies. In the former, the emphasis is put on making tacit knowledge explicit. While the latter, gives more importance to human communication leaving knowledge in its tacit form.

According to [32], being knowledge human-based, it is relevant to consider that most of the times, when needed, people ask other people about specific knowledge. For this reason, [32] proposes that the challenge for knowledge management tools is to be useful enough to be preferred by its users over human to human exchange. But at the same time, knowledge models should include information about experts so they can be referenced when complexity cannot be easily represented.

Rationale Management

Rationale in its purest meaning stands for the fundamental arguments for a reason, or in other words, the reasons behind a decision. In the research community, there is abundant literature about design rationale focusing on the design phases of several fields like software engineering, civil engineering, mechanical design, etc. [13]. Narrowing the defintion, rationale methods in software engineering have been researched as a form of capturing a type of knowledge that is never made explicit [14] by developers and contains valuable information about the software. The intent of this section is to present an overview of rationale knowledge management approaches, methodologies, and the status in the real world in preparation to the following section that will present its application to use it in ERP implementations by VARs.

DESIGN RATIONALE

Design rationale is considered a topic of research because it can yield potential benefits by capturing the reason behind a design decision. Design rationales attempt not only capture the reason behind the decision but also the explanation for it including alternatives evaluated, tradeoffs, and the knowledge process that led to the decision [15]. In [15], uses of a design rationale system are grouped according to its potential benefits: Better Design; better maintenance; learning; and documentation.

In a survey of the topic presented in [13], it was found that research to design rationale has been made since early 1980s. This research has included a wide range of publications types but the main effort is on develop prototypes and systems that by means of a representation schema help designers record their reasoning or have it done automatically by the system.

The approaches to building design rationale systems are either process-oriented or feature-oriented [13]. In the first case, the emphasis is on capturing the history of the design process and its structure is mainly descriptive. Most of the available design rationale systems are process-oriented. For the second approach, a more formal representation is used in which rules are used to evaluate the decision according to a context.

Design rationale system can also be characterized by representation schema, capture, and retrieval [13]. Representation schemas are presented ahead in this section. For capture, the classification includes either methods that require user intervention or automatic. The retrieval of knowledge rationale can be through navigation by the designer, query-based, or triggered automatically during design.

RATIONALE IN SOFTWARE ENGINEERING

The application of rationale methods has also been studied by several authors with the purpose of achieving similar objectives to the ones designers in general pursue. Reasons to include rationales as part of software engineering include: Support negotiations during requirements and facilitate elicitation of domain knowledge, structure and capture design meetings, and predict change impact [14]. For the first reason, they can be used as negotiation support to clarify reasons to help solve multiple issues raised during the requirements and design process since they involve many participants with different backgrounds. Also, by explicitly exposing the rationale, users are encouraged to review their reasoning and be part of the decision process. For the second one, a systematic approach to asses and specify options helps developers to improve the quality of their decisions along with preserving them. Finally, due to the changing characteristics of software artifacts, capturing justifications behind decisions helps maintainers to

understand interdependence and repercussions of a change avoiding *architectural erosion*.

MODELS AND REPRESENTATIONS

QOC. Questions. Options. Criteria

The QOC representation of rationale is a semi-formal notation based on a project space.

> The project space is a set of relationships or conceptual dimensions used to compare projects and alternative solutions [17].

The QOC model is composed by 4 different steps [14]:

- Questions describe the problem to be solved.
- Options represent the alternatives that were considered as answers to the question.
- Criteria include justifications used to evaluate available options by qualifying it.
- Arguments represent users' reasons to justify or oppose to the option.

IBIS. Issue Based Information Systems

Issue Based Information System (IBIS) was the first attempt to capture design rationale and it was described by Kunz and Rittel [18]. This model was not reflected in a software system but instead it was a way of modeling argumentation. IBIS captures ideas and relationships as they occur during design discussions and its main focus is on the way alternatives are extracted [17]. Issues are expressed as questions with positions that answer that issue and can be either accepted or declined. Finally arguments support or object a position [13]. Several design rationale systems have been based on this model. Some of the most recent implementations are: REMAP [19], REMAP/MM[20], KBDS-IBIS [21, 22], and IDIS [23].

DRL. Design Rationale Language.

This notation is an extension of QOC and gIBIS (hypertext-based graphical IBIS) models that include a more complete argumentation model by a language that represents the decision process. According to [24] there are 5 main entities: Decision problem, goal, alternative, claim and question. In [25] procedure was added.

- A decision problem is a controversial item in the project.
- A goal is the criteria to fix the problem.
- Alternatives represent different solutions to decision problems.
- Claims are used to argument.
- Questions lead the discussion during the project
- Procedures are steps that are followed to obtain answers.

The relationships between these entities or nods are [25]:

- A decision problem is a subdecision of another decision problem.
- A goal is a subgoal of a decision problem.
- A claim supports or denies an alternative or another claim.
- A claim answers a question.
- A question queries a claim.
- A procedure is an answering procedure for a question.
- A procedure is a subprocedure of another procedure.

Based on these relationships a knowledge-based tool called SIBYL was created to represent reasons for project decisions [26]. With this tool it is possible to manage dependency, precedence and plausibility (power of the argumentation) [17].

BARRIERS

Even though the benefits of rationale management systems are explicit and the amount of research on the topic is abundant, they have not been extensively adopted in

industry [13]. According to [13], the challenges for future developments and research are classified in representational, capture and retrieval.

Representational challenges

Representation of design rationale to be useful must have the following qualities: ease of input, effective view, and activeness. The basic goal of a successful representation is to encode knowledge in a way that can be shared and reused. For instance, in the case of design rationale systems based on IBIS, arguments are not represented as encoded text so the system cannot interpret the design.

Capture challenges

One of the key points in design rationale systems is to minimize the overhead involved in capturing knowledge without interfering with the creative process of design by working on documentation tasks [13]. Also, capture and process in real-time should be sought instead of doing it after the fact when decisions have already been made.

Retrieval challenges

The amount of knowledge that is amassed from the design process grows exponentially so adequate tools for users to retrieve it are required without having to navigate the rationale space.

RATIONALE REALITY

The ultimate goal of design rationale or rationale in general as seen by many researchers is to improve designs beyond just transmitting information between designers or developers [27]. Although when applied to software development projects it has been proved that models are successful in capturing and representing knowledge, human challenges are the ones that hinder its success [17]. According to [27], the focus should be put on supporting design processes rather than design tools and creating systems that

are more specific to the context in which they will be used. There is also agreement that useful and usable systems should be human centered and helpful rather than intrusive and learn from the knowledge management community instead of be seen as a disconnected area of investigation [13].

Rationale Management for ERP systems Value Added Resellers implementations

Unique characteristics about ERP implementations make it stand out from every other type of software engineering project. It can be argued that an ERP system is closely related to commercial off-the-shelf software packages due to the facts that it is the same system that is resold or licensed to all clients. However, what mainly differentiates an ERP from other software installations, in addition to its inherent complexity, is the level of flexibility to adapt to the client's requirements. The result is that each client, even though uses the same identical system, after the customization process they end up having a system that is unique. As discussed earlier, this characteristic represents a challenge for the VAR because it is the party that is in charge of support many different implementations. VARs not only need to be experts about the ERP they implement but also must be highly knowledgeable about each implementation and consequently have a deep understanding of each client's business.

IMPLEMENTATION'S DESIGN DECISIONS

For a VAR, an ERP implementation is a process that is characterized by knowledge intensive activities during which knowledge is elicited, transformed and applied to make the decisions that shape up the implementation. During the earlier stages in which the objectives and the vision of the project are determined, knowledge about the project is just data and facts that will guideline the projects in the next stage. Once the company has determined the need for an ERP and the high level objectives are established, then it comes the design phase. In it, strategic decisions about the implementation will be made by evaluating the differences between the list of requirements and the standard functionality of the ERP. These gaps between the as-is

business' processes and the ERP's have to be filled by one of the available levels of customization. From the simplest to the more complex they are: Parameterization, minor customization, major customization, extension.

Parameterization

Through setting parameter values, almost every software system can be adapted to the user's requirements. In the case of the ERP, this is the standard level of flexibility provided by the vendor and even though it can be very sophisticated in order to tailor the system to stick to the client's requirements, it is impossible to cover every scenario. Examples of this are: setting up the segments of customer identifications, the steps that an order has to go through to be processed, number of decimal places for quantities, etc. It is important to mention that customizations made through parameterization are fully supported by the vendor since it's a standard feature of the ERP.

Minor customization

At this level the ERP begins to separate from ordinary software systems. A minor customization involves changes to the ERP that do not affect the internal processes of the system. Examples of this kind of customization would be: Change in the name of a field in a screen or a report, add an existing field from a table in the database for user personalized information, make a calculation before a field is saved in the database, etc. This level of customizations introduces a greater level of flexibility and, even though is supported by the ERP, it could lead to scenarios under which an internal process rule could be violated.

Major customization

A major customization is one which will change an internal process the ERP implements. In general, this type of customizations will require changes to those parts of the ERP's source code that are not compiled or encrypted. Some examples are: Changing SQL code inside stored procedures to modify the way a transaction is committed, altering the standard behavior of a program or report, etc. Customizations at this level are not supported by the ERP's vendor and even though it is possible to make them, there's a risk to lose the modifications when an upgrade or fix for the ERP is applied. For VARs, each increase in the customization level represents a greater challenge that makes preserving knowledge about these modifications even more critical.

Extension

When it is not possible to accommodate a requirement through the previous customization levels, most of the VARs will be able to offer additional services to extend the functionality of the ERP. In some cases, ERPs provide with the same development platform that are used to create the programs that constitute the modules. In this case, there's the advantage that the new programs will be treated like every original program included originally in the ERP and run within its security framework. When the requested functionality is not supported by the ERP's development tools, external programs must operate outside of the ERP's environment and require an interface to integrate the data to the ERP's database. At this level, customizations are no longer treated only as part of the implementation but also as separate software development projects which have its own lifecycle. A VAR will require preserving the knowledge about the reasons why they addressed the gap through this type of customization as well as all the

internal knowledge of the development and the points of integration with the ERP.

Decisions made by consultants in this phase are the most crucial because they involve the greatest amount of applied knowledge and therefore preserved. All design decisions are influenced by negotiable and non-negotiable positions that either the client or the ERP will concede or not. In the client's case it is possible that a requirement will conflict with the way the ERP processes some information but it is critical for the organization because it represents some competitive advantage. Under this scenario, a customization to the ERP will be required to accommodate the request. On the other hand, although the ERP provides with great flexibility to be tailored, some things will not be possible to change and then is the client the one that will need to adjust its processes. In the extreme case there is no way to reconcile the differences then the best alternative is to consider another ERP that works closer to the way the client's requirements.

From the VAR's standpoint, this process requires that consultants working in an implementation not only understand the system from the back and front ends but also the client's requirement's rationale. This in order to determine the best way to design the implementation to accomplish short and long term goals from a business and technical perspective. It is evident that the decision process involves the consideration of many different factors like requirements, goals, objectives, functionality, etc. Consultants then are responsible for putting them all together and add their own expertise and previous experiences to determine what will be the solution and most importantly how to accomplish it.

KNOWLEDGE PRESERVATION OBJECTIVES

When it comes to preserving knowledge about implementation decisions, the documentation that is typically produced only captures the final decision a consultant

made. But this solely piece of knowledge is not enough or necessarily useful to achieve the objectives a VAR would pursue. Any knowledge preservation initiative has to be oriented to achieving the following objectives:

Supportability

Preserved knowledge will be used majorly in support activities as the implementation matures and after the system goes live. As it was mentioned before, on top of the ERP's implementation natural complexity, there is an added complexity factor that is directly dependent of the amount and level of customizations. Knowledge could be needed months after the design decisions were made and its consumer could either be the consultant in charge of the design decision or a different consultant that is solely responsible for support. Regardless of its consumer, in order to make good decisions, it is not enough to just read what the decision was but the reasons behind it. A support request could imply changes to the implementation design and these changes could potentially cause disruptions in the ERP operation if they violate another requirement, objective or restriction.

Education

Another objective of preserving knowledge is to pass it along to other consultants. This could be for new consultants that are learning about the ERP and how to implement it or experienced ones that are learning a new module. Also, education objectives reach towards training end-users and administrators in the way their particular ERP implementation is working and why certain decisions were made. Transferring this knowledge is a critical piece that contributes to successful ERP implementations successful since just memorizing repetitive tasks required to

operate the ERP is not enough. Taking a step further than memorization by making sure the reasons behind decisions are known makes a difference for achieving the sought after competitive advantage an ERP implementation should bring in to the company.

Conservation

As mentioned previously, it is critical for a VAR's success in its business to be able to preserve knowledge in a way that can be transferred and reused without depending on a human element. Preserving knowledge on physical artifacts like physical paper is not enough since once information is printed it becomes dead. Conserving knowledge entails preserving those key pieces that are meaningful to achieve the desired objectives. Our proposal to attain these objectives focuses on identifying the decisions consultants make that are documented as data about the implementation but without being complete knowledge. Meaningful knowledge includes the decision itself plus all the facts, information, and previous experiences a consultant used to make the decision so this can be understood and replicated when transferred.

Transferability

In order to be transferred, knowledge needs to be codified. The ultimate goal of any knowledge preservation attempt is to be able to transfer it when needed. Knowledge can be transferred through informal ways, like natural language communication between two human parties. But in order to remove one of the human elements it is required to codify it in a way that can be stored in an electronic form. In addition, the most important codification goal is to make

knowledge retrievable, otherwise the most ambitious knowledge management effort is useless.

KNOWLEDGE IN VAR'S IMPLEMENTATIONS

Characteristics

Once the objectives and potential uses described above are established, it is important to understand the characteristics of the knowledge that will be preserved in order to steer a knowledge preservation strategy. The broadness of areas an ERP implementation encompasses makes very difficult to have a single consultant that knows all the business and technical details. Requirements gathered provide with referential information about the implementation but those are not knowledge. Neither is knowledge implementation decisions documented or reflected in the value of a parameter, a required step in a process, or the segmentation of a customer identifier. So, what is knowledge in an ERP implementation?

True knowledge from an implementation has to be characterized not only by the decision made but by the process that lead to make that decision. The act of thinking can be seen as a process by which knowledge is applied to make a decision. As humans we make decisions all the time based on previous experiences, facts collected from the environment, and deductions. Putting them all together is what we call knowledge. In order to preserve it, we need to extract from the head of its owner the reasons he considered to make certain decision instead of the other thousands of possibilities. Which facts were decisive? Where did he obtain them? How they were prioritized? What implications made him discard the alternatives considered and what outcome he foresaw if they were chosen? How this decision is related to other decisions and what is their interrelation?

In summary, the vital question that needs to be answered is *why*. An answer for a *how* can be obtained from a reference manual and by memorizing. A *where* answer from searching in the right place. *What* is the outcome of the decision and is what it is perceived. But by answering *why*, knowledge is elicited and the potential of transferring it increases.

Training a user or a consultant in how to use a system, in this case, an ERP is not problematic. Making sure he repeats the task several times, guarantees he will learn how to do it. But to truly understand how to use a system effectively and in this case, the details of the implementation, requires a transfer of knowledge and of the rationale behind the decisions. This is the key piece of knowledge preservation, preserving and sharing the *whys*.

As mentioned before, consultants make the decisions during implementations to shape the ERP to satisfy client's requirements. All these decisions, and the reasons behind them are rationale and its preservation is fundamental for the client but specially for the VAR since leaving it in the head of consultants jeopardizes this asset called knowledge. The reasons are evident since as time passes consultants will work on other implementations, write more code, and eventually forget these details. But preserving rationale gets very close to obtaining the knowledge required since knowledge from the implementation can be seen as information from previous experiences, facts, and data applied to a certain situation and the justification of the expert that owns that knowledge. This way rationale can be treated as knowledge that is useful for preservation since it is dissected of its components in order to be codified, preserved, and transferred.

Types of Knowledge

During an ERP implementation we can identify three different types of knowledge.

Technical knowledge

This knowledge refers to installations and deployment procedures along with the decisions that were made. It could be seen as explicit knowledge that is found in installation manuals that have to be followed to successfully install the system. However, the environment where the system will run has many variables that require from the technical consultant's expertise and rationale to make decisions about the deployment and to overcome or prevent problems.

Implementation knowledge

As mentioned earlier, during the design phase, decisions are made about how to implement the ERP. All these can be found in design documentation but the most important pieces originate from the alternatives a consultant evaluated and why he chose them over others considered.

Customization's knowledge

Developers are responsible for creating customizations that involve writing code to customize, or extend functionality. Like with every other software development, making modifications in the future to a custom development is very likely. For this reason, preserving rationale to understand the developer's thinking when making a modification is vital to observe restrictions already considered and paths previously explored without breaking the intended functionality of a specific development or the system in general.

Codification

To be able to preserve rationale as the source of knowledge it is first required to understand what its sources are. In ERP implementations these can be classified in 3 groups:

- **Requirements.** For the implementation project, the vision, objectives, goals, and detailed requirements constitute the framework that rule a decision process.
- **Environment.** The physical and hardware environment, and the business processes are other inputs to be considered when a decision is made.
- **Architecture.** The design of the ERP's architecture is the last piece that completes the inputs of a decision.

By using these building blocks of a design decision we can design a model that considers all these inputs, considered alternatives, reasons not chosen, and the ultimate decision. Also, most importantly, each input can be codified to be able to track it and determine interrelations with other decisions flows and alternatives.

KNOWLEDGE MODEL

The proposed knowledge model, shown in Figure 1, presented in this research is based on the rationale of ERP decisions and includes the fundamental parts that constitute it. The intent is to be used as a reference for a potential system that can be used to preserve rationale including traceability from its components.

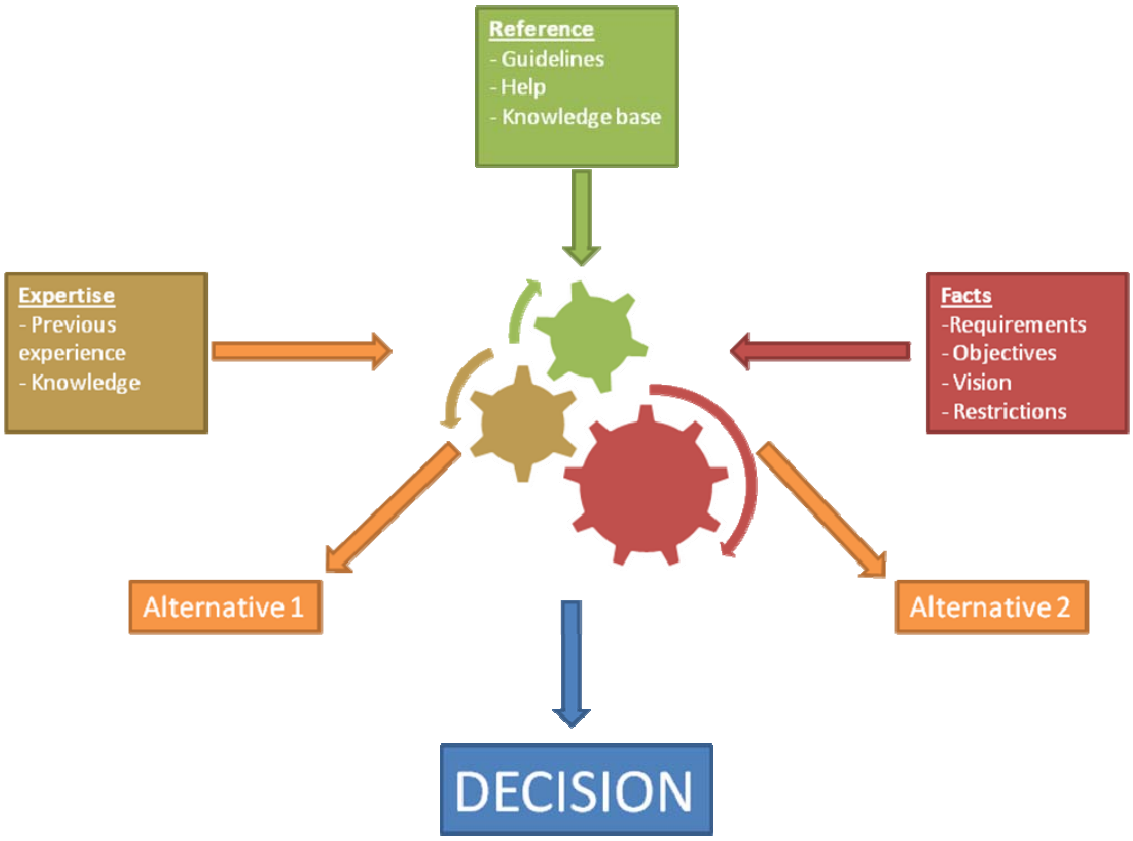

Figure 1: Rationale Model

Facts

Facts are the inputs that include the pieces of knowledge that are explicitly documented and codified therefore easily referenced when required. In a decision, they are taken into consideration as the main boundaries of which alternative is to be chosen. For ERP projects, we find this knowledge in the following artifacts: Requirements, Objectives, project vision, and restrictions.

Referential Information

Referential information consists of those inputs that have already been codified and stored. This type of information about an ERP implementation can be found in

sources like: Guidelines, manuals, help, knowledge bases. Ultimately, the goal of preserving knowledge is to transform it into this type of source.

Expertise

The last input is the key element that puts all the pieces together when a decision is made. Expertise in consultants' heads from knowledge of previous implementations is put in play along with the other two inputs to consider alternatives that conform to requirements and restrictions. If needed, referential information and knowledge explicit is examined and taken into consideration.

Alternatives

When evaluating a decision, inputs are weighed and multiple options are evaluated before selecting one. Capturing these evaluated alternatives with the reasons why they were discarded is the goal of preserving the rationale as a form of knowledge.

Decision

The decision will end up in an artifact that can be a in the form of a design decision document and eventually directly reflected as a customization.

INDUSTRY SURVEYS

To obtain the opinion from VARs about knowledge preservation and rationale management, a survey was sent to 3 executives in the industry. The companies as well as the executives' names are not disclosed.

Questionnaire A

The position of the executive that answered is Vice President, Sales and Energy Solutions with 14 years of experience in the ERP implementation field. The company profile is a Microsoft Dynamics partner that delivers solutions to mid-market and enterprise size manufacturing and distribution companies.

For executive A, the term rationale management was new since he had not heard about it before. In his own words, he considers rationale management to be *"a process of documenting the business reasons why software code should be altered from a standard state in order to perform a specific function"*.

In their implementations, the artifacts preserved include: Sales meeting notes, Pre-Sales Discovery questionnaires, Key Business Requirements, GAP analysis reports, Project Charter, Statement of Work, Master Services Agreement, Functional Design Document, Technical Design Documents, Change Orders, Functional Manuals, and End user manuals. From a knowledge preservation perspective they utilize a client portal page to store and share knowledge about projects; a revision management tools to preserve all versions of customizations created by them or the client; and finally archiving tools to preserve the programs.

In his opinion, a knowledge preservation methodology should ensure all data points are captured, as well as promote all operational steps are clearly identified so every project follows the same procedures. In a scale from 1 to 5, where 1 is unimportant and 5 is critical, he graded with a 4 the importance of knowledge preservation in his organization.

About contingency measures to prevent knowledge to be lost, he mentioned that their documentation process includes reviewing throughout and post go live the information captured to ensure it is complete and up to date.

With respect to ERP critical factors, he mentioned Project Management, Defined Scope, Detailed Requirements gathering, Best practices pertaining to development techniques, Communication, Frequent reporting to stakeholders, Executive Commitment, Continuity of consulting staff, Expertise of consultants and technology people, Defined goals, Reasonable timelines and budgets, and Industry knowledge.

Finally, he considered the following phases or operations to benefit the most from preserving knowledge: Development projects, Customizations of screens, Design, User Manuals, and Data Conversion. In addition, the drawbacks of these initiatives would be increased costs in a project, a higher degree of maintenance and rigor in projects would be required, and ensure teams of consultants follow the process.

Questionnaire B

This questionnaire was answered by the Vice-President of Technology and co-owner with 16 years of experience working with ERP implementations. The organization specializes in provide ERP, Customer Relationship Management, and Forecasting, Budgeting, and Planning solutions tailored to specific industry and functional needs.

For executive B, the term Rationale Management was completely unknown and could not define its meaning. The documentation artifacts they preserve include budgets, actual, design documents, and questionnaires.

In his organization, they do not use any kind of knowledge preservation methodology and he identified Microsoft SharePoint as a tool that they could potentially employ for this purpose but without actually doing it. In fact, they rely on *"tribal knowledge"* and recognized that a widespread access to knowledge would avoid repeating the same mistakes.

For a knowledge preservation methodology, he considered that the most important features are the effectiveness by not making daily tasks less productive. He graded in a scale from 1 to 5, where 1 is unimportant and 5 is critical, with a 3 the importance of knowledge preservation for the success of the organization. The areas or phases that would benefit the most could be sales and design because it would be good to approach a new client or prospect with some knowledge of what to do from a similar past implementation.

With respect to success critical factors for ERP implementations he mentioned experience, leadership, creativity and good communication skills.

Finally, the drawbacks from adopting a knowledge preservation initiative would be the cost and productivity hits, but the most important user acceptance and adoption.

Questionnaire C

The last questionnaire was answered by a General Manager with 10 years of experience with ERP implementations. The company profile is a software engineering organization that provides services for Microsoft Independent Software Vendors (ISV) and Microsoft Business Solutions (MBS) Channel Partners offering software development, application support, and technology consulting.

For executive C, rationale management was not unfamiliar and he defined it as a methodology that documents "what if" scenarios that can be reviewed in the future.

The artifacts preserved about implementations include everything from requirements to actual test cases. They utilize Microsoft Sure Step methodology to document by means of standard templates that are *"product agnostic"*. For knowledge preservation, he refers Microsoft SharePoint custom portals as the knowledge preservation tool using them as a central repository for knowledge retention on every implementation.

In his opinion the most important characteristics of a knowledge implementation methodology include accessibility, standard reporting, ability to capture expertise, and to be scalable. He graded with 4, in a scale from 1 to 5, where 1 is unimportant and 5 is critical, the importance of preserving knowledge for his organization.

About contingency measures they implement to protect knowledge, he recognized they do not have any at this point other than have the consultants upload documentation

to appropriate portal sites. What would be missing is the mechanism to validate this has been done.

With respect to successful ERP implementation factors, he mentioned Business Plan or Model, Planning and Strategy Communication, Project Management, Change Order Management, Execution, Delivery and Testing.

Finally the areas or phases that would benefit the most would be implementations and support since consultants could be assisted in their day to day activities when they run in an issue they haven't encountered previously and decrease the amount of time to resolution. But the most important drawback is building the infrastructure.

Conclusion

Implementation of Enterprise Systems (ES), and in particular Enterprise Resource Planning (ERP) systems, is a unique field of study in software engineering. While it is true that ERP implementations do not necessarily entail development of software, they share many of the challenges, especially the ones related to the intensive use of knowledge. During ERP implementations, one of the most important roles is played by Value Added Reseller organizations whose consultants are experts in the system as well as other activities related to the adoption process. Even though papers about this topic found for this research agree with the important role these consultants play, no one has further investigated the challenges they deal with and how to better handle them.

The value added by consultants during the implementation process is nothing else than knowledge applied. The relevant question is how to support consultants and VARs with this knowledge when they need it? First of all, even though the knowledge resides in the head of the consultant, it is not his own possession so the VAR, from an organizational perspective, has an interest in preserving this asset. Also for the consultant, preserving it is of his interest since they normally support different customers and because each implementation is unique, is not possible to expect him to retain all the particularities of the implementation in his head. Even more important, the ability to have knowledge flow between consultants and other parties guarantees that the VAR can provide the same level of support no matter who is working for the client.

Being knowledge such a critical asset, and therefore preserving it so valuable, the next question this work attempts to answer is what the best way to preserve it is? The most important decisions made by consultants during implementations are design decisions that can be found documented in different artifacts. The challenge is that by

itself the final design decision does not contain explicit knowledge, therefore the proposal of using rationale management techniques to preserve it. Rationale management has been studied not only from a software engineering perspective but from a generic approach to solve design problems for other engineering disciplines. However, it still does not permeate into the industry because the value that systems or models created for this purpose do not overcomes the amount of effort required to capture it.

From executive's answers in the questionnaires applied it is confirmed that the interest in knowledge preservation is high and its value is recognized but there is unawareness about proper techniques to do it. It is noticeable that in two cases, Microsoft SharePoint is mentioned as the preferred tool to preserve knowledge. But although, this tool can provide with repositories of knowledge artifacts, it doesn't necessarily provide with all the tools a knowledge management system include for codification, location, and retrieval.

Future research on this topic should consider that a high level of formalization in knowledge preservation discourages its use because of the overhead imposed and efforts to mimic human reasoning still are futile. In this work's context, the proposal described a framework that surrounded the description of a decision and its alternatives by the main inputs a decision about an ERP design considers, like requirements, objectives, guidelines, etc. All these inputs should be traceable so relationships between knowledge objects can be built and analyzed when knowledge is retrieved.

Appendix A

QUESTIONNAIRE

Rationale Management is a Software Engineering technique intended to capture and preserve not only documentation about a system development or implementation but also all the alternatives a designer or developer considered before making the ultimate decision.

For my Master's Report I discuss the use of Rationale Management as a technique that could be employed by VARs to preserve knowledge about ERP implementations. One of the things that I want to consider is how well this methodology could be transferred from research into the real world. For this reason I greatly appreciate your help by answering this questionnaire.

1. What is your position in the organization?
2. How many years have you been working with ERP implementations?
3. Have you heard the term Rationale Management before?
4. Please define briefly what do you think the term means?
5. What type of documentation about implementations do you preserve?
6. Do you use any kind of knowledge preservation methodology in your organization?
7. Can you name any available knowledge preservation technique or tool?
8. What do you think are the most important features of a knowledge preservation methodology and tool?
9. In a scale from 1 to 5, where 1 is unimportant and 5 is critical, how would you grade knowledge preservation is for the success of your organization.
10. What contingency measures have you implemented when knowledge from previous implementations could be potentially lost like when a consultant leaves the organization?
11. In your opinion, what are the critical factors that make an implementation successful?
12. Do you apply different approaches when implementing an ERP depending of the size of the client?
13. Do you consider that preserving knowledge would impact positively or negatively the success of implementations?
14. Which phases of an implementation or operations in your organization would benefit the most from knowledge preservation? Benefit the least? How would preserved knowledge be used?
15. What do you think would be the most important drawbacks of adopting a knowledge preservation methodology?

Glossary

ERP – Enterprise Resource Planning

ES – Enterprise System

VAR – Value Added Reseller

References

1. Moon, Y. B. (2007) Enterprise Resource Planning (ERP): a review of the literature. Int. J. Management and Enterprise Development, Vol. 4, No. 3, pp.235–264.

2. Botta-Genoulaz, V. and Millet, R-A. (2005) A classification for better use of ERP systems, Computers in Industry, Vol. 56, No. 6, pp.573–587.

3. Leknes, J. and Munkvold B, THE ROLE OF KNOWLEDGE MANAGEMENT IN ERP IMPLEMENTATION: A CASE STUDY IN AKER KVAERNER

4. Jones, M., Cline, M. and Ryan, S. Exploring knowledge sharing in ERP implementation: an organizational culture framework

5. Somers, T., Nelson, K. (2002), A taxonomy of players and activities across the ERP project lifecycle

6. R. Dolmetsch, T. Huber, E. Fleisch, H. Osterle (1998), Accelerated SAP: four case studies, Institute for Information Management, University of St. Gallen, Switzerland.

7. Al-Mudimigh, A. Zairi, M., Al-Mashari, M. (2001) ERP software implementation: an integrative framework, European Journal of Information Systems (2001) 10, 216–226

8. J. Esteves, J. Pastor, Enterprise resource planning systems research: an annotated bibliography, Communications of the Association for Information Systems, 2003, pp. 1–36.

9. M.D. Travis, ERP selection: the foundation for the enterprise, APICS Magazine 9 (6) (1999), accessed at http://www.apics.org/magazine/June99/default.html, 7 January 2000.

10. Motwani, J., Subramanian, R., Gopalakrishna, P. (2005) Critical factors for successful ERP implementation: Exploratory findings from four case studies, Computers in Industry (2005) 529–544.

11. Kremers, M., van Dissel, H., (2000) ERP system migrations, Communications of the ACM 43 (4) (2000) 52–56.

12. Davenport, T., (2000) Mission Critical: Realizing the Promise of Enterprise Systems. Harvard Business School Press, USA.

13. Regli, W. C., Hu, X., Atwood, M., Sun, W., (2000) A Survey of Design Rationale Systems: Approaches, Representation, Capture and Retrieval Engineering with Computers (2000) 16: 209–235

14. Dutoit, A., Paech, B., (2000) RATIONALE MANAGEMENT IN SOFTWARE ENGINEERING, Handbook of Software Engineering and Knowledge Engineering

15. Lee, J., (1997) Design Rationale Systems: Understanding the Issues, Intelligent Systems, vol. 12, no. 3, pp. 78-85

16. L. Bratthall, E. Johansson, B. Regnel, (2000) Is a Design Rationale Vital when Predicting Change Impact?—A Controlled Experiment on Software Architecture Evolution, In Int. Conf. on Product Focused Software Process Improvement, LNCS 1840

17. KAWAMOTO, S., ALMEIDA JR, J. (2009) A Representation Model of Design Rationale for the Design of ERP Systems. INFOCOMP (UFLA. Impresso), v. 8, p. 89-100

18. KUNZ, W., RITTEL, W. (1970) Issues as Elements of Information Systems, Working Paper No 131, University of California, Berkeley

19. Ramesh, B., Dhar, V. (1992) Supporting systems development using knowledge captured during requirements engineering. IEEE Trans. Softw. Eng., 18(6), 498–511

20. Ramesh, B., Sengupta, K. (1995) Multimedia in a design rationale decision support system. Decision Support Syst., 15(3), 181–196

21. Banares-Alcantara, R., King, J. M. P. (1997) Design support systems for process engineering iii – design rationale as a requirement for effective support. Comput. and Chem. Eng., 21(3), 263–276

22. King, J. M. P., Banares-Alcantara, R. (1997) Extending the scope and use of design rationale. Artif. Intell. for Eng. Design, Analysis and Manuf., 11(2), 155–167

23. Chung, P. W. H., Goodwin, R. (1998) An integrated approach to representing and accessing design rationale. Eng. Applic. Artif. Intell., 11(1), 149–159

24. LOURIDAS, P., LOUCOPOULOS, P., (2000) A Generic Model for Reflective Design, ACM Transactions on Software Engineering and Methodology (TOSEM), v.9 n.2, p. 199-237

25. LEE, J. S., (1990) A qualitative design management system. In P.H. Winston and S. Shellard, eds., Artificial Intelligence at MIT: Expanding Frontiers, Cambridge MA: MIT Press, p. 104-133

26. LEE, J. S., (1990) A Tool for Managing Group Decision Rationale. In Proceedings of the CSCW'90 Conference, ACM Press, New York, p. 79-92

27. Horner, J. Atwood, Michael., (2006) Design Rationale: The Rationale and the Barriers In NordiCHI '06: Proceedings of the 4th Nordic conference on Human-computer interaction (2006), pp. 341-350

28. Ramesh, B., Vasant, D., (1994) Representing and Maintaining Process knowledge for large-scale Systems Development IEEE Expert: Intelligent Systems and Their Applications archive Volume 9 Issue 2

29. Borghoff, U., Pareschi, R., (1997) Information Technology for Knowledge Management Journal of Universal Computer Science, vol. 3, no. 8 (1997), pp. 835-842

30. Studer, R., Benjamins, V., Fensel, D., (1998) Knowledge Engineering: Principles and Methods Data & Knowledge Engineering Volume 25, Issues 1-2, pp. 161-197

31. M.T. Hansen, N. Nohria, and T. Tierney, (1999) What's Your Strategy for Managing Knowledge, Harvard Business Review, March 1999, pp. 106-116

32. Kneuper, R., (2001) SUPPORTING SOFTWARE PROCESSES USING KNOWLEDGE MANAGEMENT Handbook of Software Engineering and Knowledge Engineering Vol. 2

33. Basili, V., Caldiera, G., Rombach, H., (1994) The experience factory, in Encyclopedia of Software Engineering, ed. J. J. Marciniak, John Wiley, 1994, pp. 469-476

34. Rodriguez-Elias, O., Martinez-Garcia, A., Vizcaino, A., Favela, J., Piattini, M., (2009) MODELING AND ANALYSIS OF KNOWLEDGE FLOWS IN SOFTWARE PROCESSES THROUGH THE EXTENSION OF THE SOFTWARE PROCESS ENGINEERING METAMODEL, International Journal of Software Engineering and Knowledge Engineering Vol. 19, No. 2 (2009) pp. 185–211

35. Davenport, T., Prusak, L., (1998) Working Knowledge. How Organizations Manage What They Know, Harvard Business School Press, Boston

36. Bjørnson, F., Dingsøyr, T., (2008) Knowledge management in software engineering: A systematic review of studied concepts, findings and research methods used, Information and Software Technology

37. Potts, C., Bruns, G., (1988) Recording the Reasons for Design Decisions, ICSE '88 Proceedings of the 10th international conference on Software engineering

38. Chan, E., Walker, D., Mills, A., (2009) Using a KM framework to evaluate an ERP system implementation, Journal of Knowledge Management VOL. 13 NO. 2 2009, pp. 93-109

Vita

Miguel Angel Otero Lanuza was born in San Luis Potosi, Mexico to Miguel Angel Otero Narvaez and Maria del Socorro Lanuza Chavez. He graduated from Insituto Tecnologico y de Estudios Superiores de Monterrey in Leon, Mexico and received a Bachelor of Science degree in Computer Systems Engineering in 1999.

After graduation Miguel worked as an independent contractor developing software integrated to Enterprise Resource Planning Systems and designing and developing web applications.

In 2007 Miguel moved to Houston, TX to work for Nextec Group, a Microsoft Gold Partner reseller, where he customizes and develops applications integrated to commercial Enterprise Resource Planning Systems.

Miguel lives in The Woodlands, TX a suburb of Houston.

Permanent email: al561914@yahoo.com

This report was typed by Miguel Angel Otero Lanuza.